REMARKS

Claims 1-7 and 21-28 were pending. Claims 1, 5, 21, 26 and 27 have been amended for clarification purposes. Accordingly, claims 1-7 and 21-28 remain pending.

In the present Office Action, claims 1-7 and 21-28 stand rejected under 35 U.S.C. § 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. The Examiner suggests it is not clear what the communication stack is and how the stack levels are managed. Applicant has amended the claims to use the terminology "storage management stack" which appears in original claim 21 for all of the claims. Applicant offers the following comments and excerpts from the Description in order to further clarify the nature of the claimed invention.

Generally speaking, the presently claimed invention is directed to accessing storage objects in a computing system. As noted in the Description:

> "Furthermore, as used herein a computing device includes one or more processing elements coupled with computer readable memory which can be volatile or nonvolatile memory or any combination thereof. Additionally, the term "object" of "storage object" as used herein includes data storage elements such as, and by way of example only electronic files, portions of data related to a single electronic file, a file system, a database, a storage device partition, and the like." (page 9, lines 7-12).

However, with respect to the prior art, the Description notes the following:

> "SAN technology has not produced advances in throughput of operations as one would anticipate. This is due to the fact that shared access to data among several compute platforms must be mediated by distributed file systems. . . . Consequently, application writers are motivated to find strategies other than distributed file system in order to share data at speeds that are consistent with SAN technology. . . .

> For these strategies to succeed, applications need to be able to discover sufficient information about files and volumes that a component on another platform can access the data associated with the file or volume. Customarily, this type of information is not externalized by either file systems or distributed file systems.
>
> . . .
>
> Additionally, even with persistent and stable representations of data storage elements, workable and useable application programming interfaces (APIs) will need to be established, such that different levels of abstraction and interfacing to the storage elements can be achieved seamlessly with user-defined software applications. In this way, user-defined software applications can utilize the APIs to better interact with the storage objects. Moreover, the user-defined software applications will often reside in storage environments different from the storage elements, therefore any provided API must be capable of operating in both storage environments.
>
> . . .
>
> Moreover, user-defined applications are implemented at different levels of abstraction and thus a user-defined application can make reference to a storage element at different abstraction levels within the storage environment. As a result, any provided API must be capable of determining the storage element's reference level within the storage environment. Therefore, access to the private information or absolute location of the storage element presents a number of challenges when providing APIs to user-defined software applications.
>
> Therefore, what is needed are methods and systems for providing access to data storage elements residing in any storage environment, regardless of any reference level within which access is attempted on the storage element, thereby resulting in access to data storage elements becoming more seamless, transparent, flexible, and reliable across storage environments." (page 3, line 24 – page 6, line 15).

Consequently, multiple levels of abstraction and multi-platform computing present challenges in accessing storage elements. To this end, a storage management stack is configured in such a way that access to a storage objects absolute location may be obtained in a relatively transparent manner.

> "[E]ach layer of the storage management stack is responsible for hiding some level of complexity from the layer above. Moreover, each element of the stack is responsible for exporting its abstraction to the layer above it within the stack. Accordingly, the storage management stack and the elements of the stack include a number of operating system and file system services which permit the abstraction and translations to occur.
>
> To decompose a storage object within a storage management stack, the storage object reference must be translated into one or more absolute extents from the initial reference which is often at different stack abstraction level and is typically associated with one or more relative extents for the storage object. For example, some applications may reference a storage object without the need for a file system, and some file systems do not include a VM. A storage object is definable by using extents, wherein each extent identifies a storage management stack level, a beginning location within the identified stack level, and a data length associated with the extent. Furthermore, a single storage object can include several non-contiguous extents which when assembled describe all the data associated with a storage object.
>
> As services are used within the storage management stack, relative extents are translated into additional relative extents until at the lowest level, device extents or absolute extents are acquired. The absolute extents provide the exact physical locations for the one or more storage devices which house the storage object. The process of traversing the stack is referred to herein as "mapping through the stack." (page 14, line 12 – page 15, line 5).

In view of the above discussion, the stack and its use as recited in the claims becomes clear. The storage management stack facilitates access to a storage object in a system wherein the requestor does not necessarily know the absolute location of the storage object. For example, claim 1 recites a method for resolving a storage object's absolute location within a first storage environment to grant access to the storage object, comprising: receiving a storage object reference; determining an initial storage management stack level associated with the storage reference; iterating through one or more additional storage management stack levels beginning with the initial stack level in response to determining the storage reference is not an absolute reference; and translating the storage reference through each iteration into one or more relative extents until one or

more absolute extents are obtained, wherein the one or more absolute extents comprise the storage object's absolute location within the first storage environment. In this manner, various levels of abstraction, and potentially private information, may be traversed in a transparent manner to gain access to the storage object.

In view of the above discussion and the context of the Description Applicant believes the claims to be clear and in accordance with 35 U.S.C. § 112. Should the examiner believe there to be issues remaining that would prevent the present application from proceeding to allowance, the below signed representative requests a telephone call at (512) 853-8866 in order to facilitate resolution.

CONCLUSION

Applicant submits the application is in condition for allowance, and an early notice to that effect is requested.

If any extensions of time (under 37 C.F.R. § 1.136) are necessary to prevent the above referenced application(s) from becoming abandoned, Applicant(s) hereby petition for such extensions. If any fees are due, the Commissioner is authorized to charge said fees to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5760-17100/RDR.

Also enclosed herewith are the following items:

☒ Return Receipt Postcard

☐ Other:

Respectfully submitted,

Rory D. Rankin
Reg. No. 47,884
ATTORNEY FOR APPLICANT(S)

Meyertons, Hood, Kivlin,
Kowert, & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8800

Date: Nov. 30, 2004